CSC 282 - Fall 2010 - `http://www.cs.rochester.edu/~stefanko/Teaching/10CS282/`

QUIZ #4 - this quiz contributes 11% to your grade.

110 points total

1. (10 points) We are given n positive numbers $a_1, \ldots, a_n$. A selection S of the numbers is called **valid** if no two consecutive numbers are selected in S. The goal is to find a valid selection of the numbers with the maximal sum. We are going to give an $O(n)$-time dynamic programming algorithm for the problem. We will have a table $P[1..n]$ where

$$P[i] = \text{the maximal sum of a valid selection of the first } i \text{ numbers.}$$

Give an expression (or a piece of code) for $P[i]$ in terms of $P[i-2]$, $P[i-1]$, and a_i.

$$P[i] =$$

2. (20 points) A **shuffle** of two strings A, B is formed by interspersing the characters into a new string, keeping the characters of A and B in the same order (for example, '**sev**e**ral**' is a shuffle of 'seal' and 'evr'). Given three strings $A = a_1 \ldots a_n$, $B = b_1 \ldots b_m$, and $C = c_1 \ldots c_{m+n}$, we would like to verify whether C is a shuffle of A and B. We are going to solve the problem using dynamic programming. We will compute a table $T[1..n, 1..m]$, where

$$T[i,j] = \begin{cases} \text{true} & \text{if } c_1 \ldots c_{i+j} \text{ is a shuffle of } a_1 \ldots a_i \text{ and } b_1 \ldots b_j, \\ \text{false} & \text{otherwise.} \end{cases}$$

Give an expression (or a piece of code) for $T[i,j]$ (in terms of previously computed values of T and elements of A, B, C).

$$T[i,j] =$$

3. (15 points) Let $X = x_1, \ldots, x_m$, $Y = y_1, \ldots, y_n$, and $Z = z_1, \ldots, z_\ell$ be three sequences. Our goal is to compute the length of the longest common subsequence of X, Y, and Z. We will have a 3-dimensional table $T[0..m, 0...n, 0..\ell]$, where $T[i, j, k]$ will be the length of the longest common subsequence of $x_1, \ldots, x_i$, $y_1, \ldots, y_j$, and $z_1, \ldots, z_k$. Give an expression (or a piece of code) to compute $T[i, j, k]$ from the previously computed values in T.

4. (10 points) In the KNAPSACK PROBLEM we have n items. The weight of the i-th item is Q_i and the value of the i-th item is Z_i. Assume that the Q_i's are small integers. Let C be the capacity of the knapsack (assume that C is an integer). We will compute an array $K[0..C, 0..n]$, where

   $$K[B, i] = \text{the maximal total value of a subset of items } \{1, \ldots, i\} \text{ with total weight at most } B.$$

   Give an expression (or a piece of code) for $K[B, i]$ in terms of some of the $K[?, i-1]$ (the question mark should be replaced by appropriate expressions).

   $$K[v, i] =$$

5. (30 points) Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$ be two sequences of numbers. We would like to find **the longest common increasing subsequence** of $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$. Give a *dynamic programming algorithm* for the problem. Clearly and succinctly describe: 1) the table are you using, 2) how do you recover the final answer from the table, and 3) the update rule.

6. (25 points) We are given a collection of n intervals $I_1, \ldots, I_n$. Each interval $I_i = [a_i, b_i]$ is assigned some weight w_i. We want to find the max-weight subset of disjoint intervals, i. e., we want to find $S \subseteq \{1, \ldots, n\}$ such that

   - for any distinct $j, k \in S$ the intervals I_j and I_k are disjoint, and
   - the $\sum_{j \in S} w_j$ is maximized.

   Give an $O(n^2)$-time *dynamic programming algorithm* for this problem. Clearly and succinctly describe: 1) the table are you using, 2) how do you recover the final answer from the table, and 3) the update rule.

   (To simplify the problem assume that the a_i and b_i are all distinct.)